TracNav

- **Quick Links**
  - Overview
  - CQoS
  - Bocca
  - CCA Tools Installation
  - TOPS Components
  - Publications
  - People
- **See also**
  - TracGuide
  - TracWelcome

A Contractor-Based CCA Tools Build System

This project provides a package management system for the Common Component Architecture scientific computing tools, which consist of around a dozen packages. The most up-to-date CCA tools versions are available here.

This build system is based on Contractor by James Amundson, a tool written in Python for the high-level configuration and installation of multiple inter-dependent packages. The new system is modular, easy to read and easy to update, and implemented using an object-oriented approach that simplifies the process of obtaining, configuring, and installing large software collections, with nontrivial internal dependencies. Contractor supports different types of configuration and build systems, invoking underlying Autotools-based build systems, as well as enabling extensions for different build systems or any necessary pre- or post-processing. All output is logged and on failure the logs are archived to be sent to the developers for debugging. A GTK+ graphical user interface is provided to ease configuration.

Here is an early presentation on the CCA Tools installer:

- Download PDF
- Download MOV (Video version, click to move forward)

Getting the CCA Contractor-Based Build System

Latest release:

- Tarball: http://www.cca-forum.org/download/cca-tools/cca-tools-latest/cca-tools-installer-0.2.7.tar.gz
- MD5 sum: http://www.cca-forum.org/download/cca-tools/cca-tools-latest/cca-tools-installer-0.2.7.md5sum

Nightly tarballs are available:

- Tarball: http://www.cca-forum.org/download/cca-tools/nightly/cca-tools-installer.tar.gz
- MD5 sum: http://www.cca-forum.org/download/cca-tools/nightly/cca-tools-installer.md5sum

The repository can be checked out with:

```
svn co https://cca-forum.svn.sourceforge.net/svnroot/cca-forum/cca-tools-installer/trunk cca-tools-inst
```

If you are interested in contributing to our development, please email Boyana Norris.

Bug Reports and Feature Requests

To submit bug reports or feature request, email the CCA Help Desk.

Usage

Using the new contractor-based build system is very easy. Instead of the usual 'configure, make, install' scenario you now have only two steps (one if you let the system automatically configure, in which case you can omit the first step).

```
$ cd cca-tools-installer
$ ./contract.py --configure
...
$ ./contract.py
...
```

For more control over the build, you can set your own configuration options manually to override the defaults. To see the list of available options, pass the --configure-show argument, which also shows you the arguments passed to configure on the last invokation.

```
$ ./contract.py --configure-show
...
$ ./contract.py --configure python=/sw/bin/python2.5 f90=ifort f77=
...
```

As seen above, to unset an option just leave the right side of the assignment blank. Alternatively, you can use the syntax option_name=None to disable an option. The example above sets the path to the Python interpreter to use in Babel and Ccaffeine, specifies that Intel compilers should be used for Fortran 90, and disables Fortran 77 support.

Specific stages can be built easily using the new build system. There are a couple of special stages of note: the default target (called when no arguments are passed), the global clean target (to clean everything), and the per-package all and clean targets.

```
$ ./contract.py babel/make
...
$ ./contract.py babel/all
...
$ ./contract.py clean
...
```

Example Configurations

Here are some sample configurations used different architectures.

- Linux Ubuntu 8.04, dual quad-core Opteron; build latest release distributions, enable mpi, python, fortran (adding nightly=True would build from the nightly tarballs of all tools except Babel, producing this sample output):

```
../contract.py --configure python=/usr/bin/python \
                           f90=ifort f77=ifort \
                           mpi=$HOME/software/linux-Ubuntu_7.04-x86_64/openmpi-1.2.3 \
                           prefix=/path/to/install/dir
```

- MacOS X 10.4, MacBook Pro; use nightly tarball of software, specify python explicitly, enable mpi:

```
../contract.py --configure nightly=True \
                           prefix=/path/to/install/dir \
                           f77=/usr/local/bin/gfortran python=python2.3 \
                           mpi=/usr/local/openmpi-1.1.4
```

- MacOS X 10.4, MacBook Pro; use latest official releases of tools, enable mpi, use gfortran for f77

```
../contract.py --configure mpi=/usr/local/openmpi-1.1.4 \
                           python=python2.3 \
                           f77=/usr/local/bin/gfortran \
                           prefix=/path/to/install/dir
```

The Graphical Interface

The graphical user interface is provided to ease the configuration and building of the CCA tools. It is written in python and uses GTK+. To run it you must have python and pygtk with glade support enabled.

```
$ ./gui/cca-install-gtk.py
```

Screenshots

Language options setup Installation